

# BACCALAURÉAT

SESSION 2024

---

Épreuve de l'enseignement de spécialité

## NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

---

Sujet n°10

---

DURÉE DE L'ÉPREUVE : 1 heure

Le sujet comporte 4 pages numérotées de 1 / 4 à 4 / 4  
Dès que le sujet vous est remis, assurez-vous qu'il est complet.

*Le candidat doit traiter les 2 exercices.*

## EXERCICE 1 (10 points)

Dans cet exercice, on cherche à calculer la moyenne pondérée d'un élève dans une matière donnée. Chaque note est associée à un coefficient qui la pondère.

Par exemple, si ses notes sont : 14 avec coefficient 3, 12 avec coefficient 1 et 16 avec coefficient 2, sa moyenne pondérée sera donnée par

$$\frac{14 \times 3 + 12 \times 1 + 16 \times 2}{3 + 1 + 2} = 14,333\dots$$

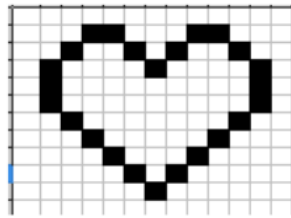
Écrire une fonction moyenne :

- qui prend en paramètre une liste notes non vide de tuples à deux éléments entiers de la forme (note, coefficient) (int ou float) positifs ou nuls ;
- et qui renvoie la moyenne pondérée des notes de la liste sous forme de flottant si la somme des coefficients est non nulle, None sinon.

Exemple :

```
>>> moyenne([(8, 2), (12, 0), (13.5, 1), (5, 0.5)])
9.142857142857142
>>> moyenne([(3, 0), (5, 0)])
None
```

## EXERCICE 2 (10 points)



On travaille sur des dessins en noir et blanc obtenu à partir de pixels noirs et blancs : La figure « cœur » ci-dessus va servir d'exemple. On la représente par une grille de nombres, c'est-à-dire par une liste composée de sous-listes de même longueur. Chaque sous-liste représentera donc une ligne du dessin.

Dans le code ci-dessous, la fonction `affiche` permet d'afficher le dessin. Les pixels noirs (1 dans la grille) seront représentés par le caractère `'*'` et les pixels blancs (0 dans la grille) par des espaces.

La fonction `liste_zoom` prend en arguments une liste `liste_depart` et un entier `k`. Elle renvoie une liste où chaque élément de `liste_depart` est dupliqué `k` fois.

La fonction `dessin_zoom` prend en argument une grille `grille` et renvoie une nouvelle grille où toutes les lignes de `grille` sont zoomées `k` fois et répétées `k` fois.

Compléter les fonctions `liste_zoom` et `dessin_zoom` du code suivant :

```
def affiche(dessin):
    ''' affichage d'une grille : les 1 sont représentés par
        des "*" , les 0 par un espace " " '''
    for ligne in dessin:
        affichage = ''
        for col in ligne:
            if col == 1:
                affichage = affichage + "*"
            else:
                affichage = affichage + " "
        print(affichage)

def liste_zoom(liste_depart,k):
    '''renvoie une liste contenant k fois chaque élément de
        liste_depart'''
    liste_zoomee = ...
    for elt in ... :
        for i in range(k):
            ...
    return liste_zoomee
```

```

def dessin_zoom(grille,k):
    '''renvoie une grille où les lignes sont zoomées k fois
    ET répétées k fois'''
    grille_zoomee=[]
    for ligne in grille:
        ligne_zoomee = ...
        for i in range(k):
            ... .append(...)
    return grille_zoomee

```

Exemples :

```

>>> coeur = [[0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0],
              [0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0],
              [0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0],
              [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
              [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
              [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
              [0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0],
              [0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0],
              [0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0],
              [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]]
>>> affiche(coeur)

```

```

** **
 * * * *
*      *
*          *
*              *
 *          *
 *      *
 *      *
 *      *
 *

```

```

>>> affiche(dessin_zoom(coeur,2))

```

```

****      ****
****      ****
**      ** **      **
**      ** **      **
**          **      **
**          **      **
**          **      **
**          **      **
**          **      **
**          **
**          **
**          **
**          **
**          **
**          **
**          **
**          **
**          **
**          **
**          **
**          **
**          **
**          **
**          **
**          **
**          **
**          **
**          **
**          **
**          **

```

```

>>> liste_zoom([1,2,3],3)
[1, 1, 1, 2, 2, 2, 3, 3, 3]

```